FOSSpicks

Sparkling gems and new releases from the world of Free and Open Source Software

Mike Saunders has spent a decade mining the internet for free software treasures. Here's the result of his latest haul…

Partition manager

GParted 0.20

Version numbers are funny things. Some developers would argue that they don't mean much, but end users often infer a lot from them. We know of quite a few FOSS programs that have surprisingly low version numbers, given their quality and feature set, such as *Inkscape*. It has been around for years, has loads of features, is used by professionals for real work, and yet it's only at 0.48. The same could be said of *GParted*: it's a hugely useful and mature tool, yet its 0.20 version number suggests it's barely at alpha stage.

That might scare some users away, especially given its job of performing (potentially risky) filesystem operations. In our experience over the years, though, *GParted* is a tool you can rely on. It's a *GTK*-based program for managing partitions on your hard drive, and it's available in two flavours: as a standalone app, or as a live distro. The latter is what we're looking at here, as it's very useful to have around on a spare CD-R or USB key for emergencies.

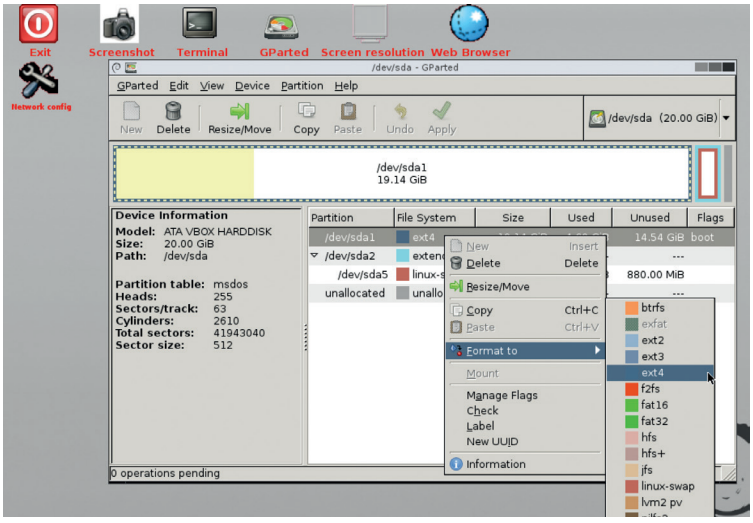
The ISO weighs in at a touch under 200MB, and when booted, it asks for your keyboard layout and language. Then it drops you into a rather ugly Fluxbox desktop, with some dreadful desktop icons. Sure, glitz and fancy effects aren't important here, but a bit of work on the presentation would be welcome, especially when you're using it to fix someone else's system.

Anyway, *GParted* itself pops up, showing the layout of your hard drive. (If you have multiple drives, you can select the one you want to edit via the drop-down menu on the top-right of the window.) Click on a partition in the list to select it, and then use the toolbar at the top (or the right-click menu) to perform an operation on it. Note that *GParted* doesn't perform its operations straight away – instead, it batches them up until you click the Apply button at the top.

GParted is tremendously versatile, supporting over 20 filesystem formats, although not every operation is available for each format. But for the common formats (ext*, btrfs, NTFS, FAT32, HFS+) you can create, copy, resize and relabel partitions. In many cases you can also perform checks on partitions, and attempt to recover deleted ones. It's a great toolbox for working with hard drives, and an essential part of a sysadmin's armoury.

This panel shows the range of filesystems that are supported, and operations available for them.

> **"GParted is very useful to have around on a spare CD-R or USB key for emergencies."**



The GParted Live distro isn't a feast for the eyes and looks very late 90s, but it gets the job done.

PROJECT WEBSITE
www.gparted.org

Operating system

Minix

Although it's not very well known, Minix played a major role in the early days of Linux. It was created by computer science professor Andrew Tanenbaum as a learning tool, and Linus Torvalds used it to build the very early releases of his kernel. The two hackers got into a fascinating online debate about kernel design – a debate that has become so famous, it even has its own Wikipedia page: **http://tinyurl.com/b2us8t**. Tanenbaum argued that microkernels are the future, and Linux was already obsolete before it had even taken off. Torvalds disagreed, of course.

But what is a microkernel? Essentially, it's a very small kernel that does a handful of vital jobs: mapping memory, controlling processes, and enabling processes to communicate with one another. Everything else, including hardware drivers, networking protocols and so forth, are run in "userspace", where they can't interfere with the inner workings of the kernel. Contrast this to Linux, where a good chunk of the OS's functionality is provided directly inside the kernel.

So a microkernel should, in theory, be more reliable, as parts can be swapped out more easily in the event of a bug wreaking havoc. But when you have more things happening in userspace, and major parts of the system exchanging messages all the time, performance can be impacted significantly. So the debates continue today, but Minix 3 is a good example of a genuinely useful all-round OS built on a microkernel design.

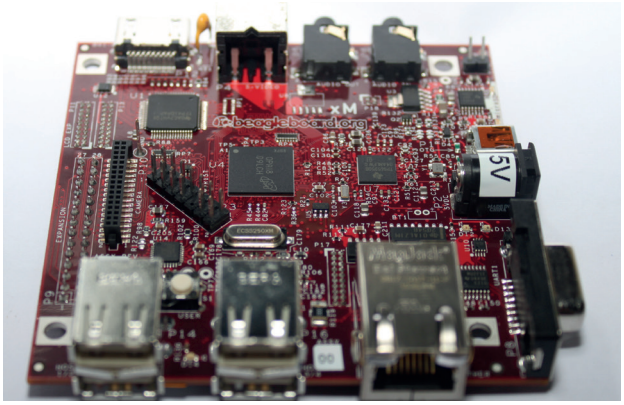
A bit of this, a bit of that

Minix 3 is now regarded as an alternative to Linux and the BSDs for embedded devices and low-spec hardware and it runs many FOSS apps you're familiar with. To install it, you'll need at least 64MB of RAM and an i586 or newer processor along with 675MB of drive space.

Minix 3.3 is provided as a 288MB compressed ISO image, which extracts to 578MB, so it can be burned to a regular CD-R. After booting, you'll be prompted to log in as root (without a password), and then run **setup** to begin installation. This is all plain text, and there are no hand-holding wizards, but if you have some Unix experience you won't find it too daunting.

By and large, Minix has a familiar Unix-like userland, which isn't surprising as much of it has been taken from NetBSD in recent releases. Indeed, Minix also uses NetBSD's *Pkgsrc* system, so a wide range of software is available with just a few commands. However, X wasn't working at the time of writing – the developers were still in the process of moving away from the crusty old XFree86 codebase.

Ultimately, Minix feels a lot like Slackware and Debian from the mid-90s. It's surprisingly usable and interesting to explore, and the documentation isn't bad either. It won't be challenging Linux or FreeBSD any time soon, but if you're interested in exploring alternative kernels or just want to expand your Unix knowledge, give it a go in *VirtualBox* or *Qemu*. (Note: for the former, create a virtual IDE hard drive image.)

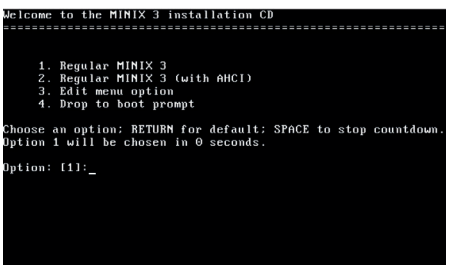


Minix 3 is the first release to support ARM chips, and specifically BeagleBoards (image: Mapper 07, Wikipedia).

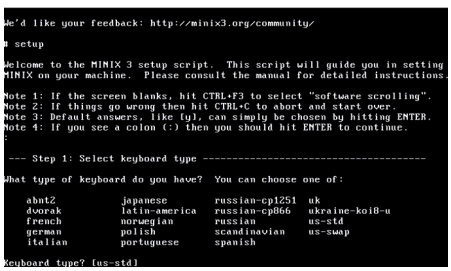
"Minix 3 is now regarded as an alternative to Linux and the BSDs for embedded devices."

PROJECT WEBSITE
www.minix3.org

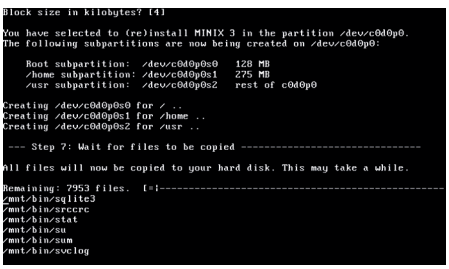
How it works: Installing Minix3


Welcome to the MINIX 3 installation CD
1. Regular MINIX 3
2. Regular MINIX 3 (with AHCI)
3. Edit menu option
4. Drop to boot prompt
Choose an option: RETURN for default; SPACE to stop countdown.
Option 1 will be chosen in 0 seconds.
Option: [1]:

1 Boot
You can use a real PC, but it's easier to just use *VirtualBox* or *Qemu* to boot the ISO image – and at this menu, hit Enter to boot up Minix.


We'd like your feedback: http://minix3.org/community/
setup
Welcome to the MINIX 3 setup script. This script will guide you in setting
MINIX on your machine. Please consult the manual for detailed instructions.
Note 1: If the screen blanks, hit CTRL+F3 to select "software scrolling".
Note 2: If things go wrong then hit CTRL+C to abort and start over.
Note 3: Default answers, like (y), can simply be chosen by hitting ENTER.
Note 4: If you see a colon (:) then you should hit ENTER to continue.
--- Step 1: Select keyboard type ---
What type of keyboard do you have? You can choose one of:
abnt2 japanese russian-cp1251 uk
dvorak latin-america russian-cp866 ukraine-koi8-u
french norwegian russian us-std
german polish scandinavian us-swap
italian portuguese spanish
Keyboard type? [us-std]

2 Install
Log in as root and enter setup to begin the process. You'll be prompted for your keyboard layout, and then move on to drive partitioning.


Block size in kilobytes? [4]
You have selected to (re)install MINIX 3 in the partition /dev/c0d0p0.
The following subpartitions are now being created on /dev/c0d0p0:
Root subpartition: /dev/c0d0p0s0 128 MB
/home subpartition: /dev/c0d0p0s1 275 MB
/usr subpartition: /dev/c0d0p0s2 rest of c0d0p0
Creating /dev/c0d0p0s0 for / ..
Creating /dev/c0d0p0s1 for /home ..
Creating /dev/c0d0p0s2 for /usr ..
--- Step 7: Wait for files to be copied ---
All files will now be copied to your hard disk. This may take a while.
Remaining: 2953 files. [=]
/mnt/bin/sqlite3
/mnt/bin/srccrc
/mnt/bin/stat
/mnt/bin/su
/mnt/bin/sum
/mnt/bin/svclog

3 Wait
If you can devote the whole drive to Minix, use the automatic mode. The files will be copied over, and then you can reboot into your new Minix installation.

Old school terminal emulator

Cool-retro-term

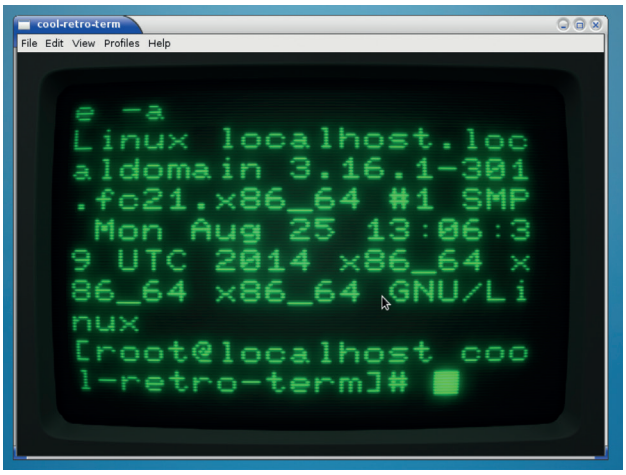
Hollywood typically depicts "hackers" as bespectacled geeks sitting in front of green text terminals with all sorts of incomprehensible gobbledygook scrolling by. They'll use these cliches even in films set in the current decade, despite how daft they are. But go back to the 70s and 80s and you would actually find these flickery, eye-strain-inducing displays hooked up to mainframes and minicomputers.

If you want something today that really harks back to the good old days of Unix, try *Cool-retro-term*. Its name says it all: it recreates an old-school terminal on your desktop, and it's pretty cool.

Well, apart from compiling it. The developers have provided a helpful list of dependencies on the project's website, but if you're not running KDE you'll be pulling in a giant bunch of extra packages. The interface is built around *Qt 5.2*, so you'll need an up-to-date distro to build the source code – we used a test release of Fedora 21. We followed the instructions to the letter, but we still had to install some extras manually.

Anyway, once it's running, it's pretty awesome. The developers have done a great job of recreating an old-school text terminal, beyond just making everything pixellated to the extreme. There's a fuzziness around the characters, irregularity in the lighting, twitches from the virtual CRT, and warping at the edges (simulating the bulge of a non-flat screen). Even if you never used a Unix box in the 70s, it may still remind you of 8-bit computers hooked up to 80s TV sets.

Unfortunately, we couldn't access any of the settings; the menus simply didn't work. It's not clear whether this is due to a bug in the program or the mid-development status of Fedora 21, but we couldn't explore some of the extra features. Still: you can normally customise the font size and colour scheme, and switch into a full-screen mode.



Top tip: enable this full-screen on your laptop, go into a coffee shop, and watch as everyone assumes you're the l33t3st h4x0r in the world.

PROJECT WEBSITE
https://github.com/Swordfish90/cool-retro-term

Screenshot taker

Maim

One of the best command line tools for taking screenshots is *Scrot*. With a command line tool you can take batches of screenshots, selecting specific areas, which is very handy when you need to do repetitive jobs (such as collecting images for software documentation). *Maim* claims improve on *Scrot* with some features that the former lacks.

The main dependencies are **lmlib2**, **libXrandr** and **libXfixes**; these are available as **libimlib2-dev**, **libxrandr-dev** and **libxfixes-dev** on Debian-based distributions. With those in place, grab the latest source code and build it like so:

```
git clone https://github.com/naelstrof/maim.git
cd maim
cmake ./
make && sudo make install
```

You can use the program straight away by entering, for example, **maim foo.png** to save the whole screen as **foo.png**. You'll probably want to get the terminal window out of the way first, though, so add **-d** followed by a number to delay the screenshot-taking process for the specified number of seconds.

If you want to grab a part of the screen, you can specify coordinates using the **-x**, **-y**, **-h** and **-w** flags; in many cases, though, you'll want to select an area yourself. *Maim* can't do this on its own, but if you install *Slop* (linked to on the project's website) then the **-s** flag lets you either click and drag to select an area, or click on a window's titlebar to take a screenshot of just that window. The selected window is highlighted with a grey border, which you can customise using extra settings. It's also possible to select a specific window using the **-i** flag. This requires the window ID, which you can get by running **xdotool selectwindow**, clicking on the window you want to capture, and noting the number. Enter **maim --help** for the full list of options.

Maim makes off-screen areas transparent, which is useful if you have multiple monitors with different resolutions.

PROJECT WEBSITE
https://github.com/naelstrof/maim

Atari-like virtual machine

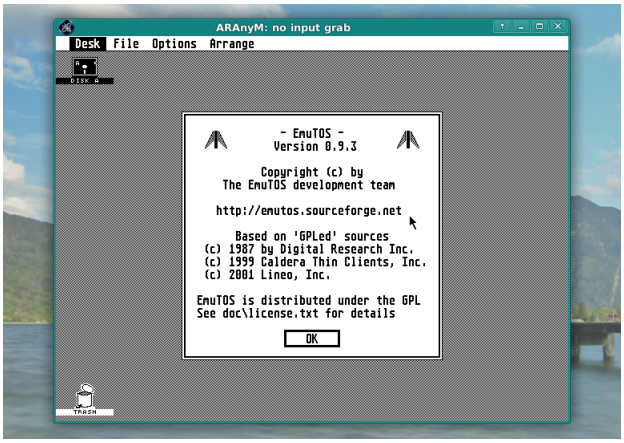
Aranym 0.9.16

There are many Atari ST/TT/Falcon emulators doing the rounds, and most of them have very good compatibility with the original machines. *Aranym* (a contraction of "Atari running on any machine") is slightly different in that it doesn't emulate a specific model from the ST series; it just emulates the best hardware suitable for an open source version of TOS, the ST's old operating system.

After you've installed *Aranym*, you'll also need to get a TOS image from **http://emutos.sf.net**. Like the original TOS, EmuTOS is a single-tasking operating system with certain limits, but it's completely open source and works really well with *Aranym*. Fire up *Aranym* for the first time, and then close it; you'll now have a file called **.aranym/config** in your home directory. Edit this to point the EmuTOS line to the location of the file you downloaded (eg **/home/mike/etos512k.img**) and start *Aranym* again. All being well, you'll arrive at the desktop.

From here, you can go about setting up your virtual Atari as you wish. You can reedit the configuration file to enable hard drives using disk images or directories, and also tweak the video settings as well. Note that *Aranym* completely takes over your mouse input when you're using it, so if you need to get the cursor back for other Linuxy work, press left Shift, left Control, left Alt and Escape at the same time.

Aranym is compatible with many original ST/Falcon games and apps, and you can also enable networking and get it online. Even if you didn't have one of the machines at the time, it's still great fun to play around with an OS that's half a relic of yesteryear, and half being kept alive thanks to the passion of open source hackers.



Aranym features JIT (just-in-time) translation for the emulated CPU, so it runs lightning fast.

"Aranym is a contraction of 'Atari running on any machine'."

PROJECT WEBSITE
http://aranym.sf.net

Text-mode Twitter client

Rainbowstream

A command line Twitter client may seem like a crazy idea, but in most cases it works perfectly well. After all, tweets are really just plain text, and if you've ever spent much time with *Mutt*, *WeeChat* and similar tools, you'll know that text-mode programs can be faster and more efficient than their GUI equivalents. *Rainbowstream* is written in Python, so the quickest way to get it is:

```
pip3 install rainbowstream
```

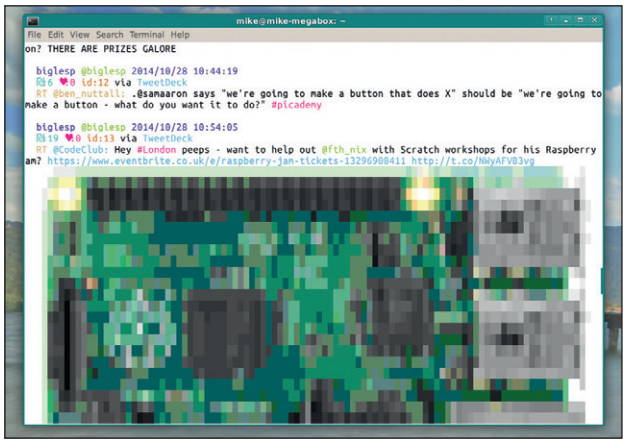
The main dependency is *Pillow*, a fork of the Python Imaging Library. Once you have it installed, enter **rainbowstream** and your web browser will pop up, prompting you to authenticate on Twitter. What's going on here? Well, Twitter doesn't let random applications use your account, so first it performs a check, confirming that you want to let *Rainbowstream* read and post to your feed.

Back in the terminal, you can now begin using Twitter. Enter **h** to display a list of help topics, which you can then follow with another word – for instance, **h tweets** will show you how to compose new tweets. Entering **home** (optionally followed by a number) will show the most recent tweets on your timeline, while **mentions** will show tweets that mention you. It doesn't take long to master, and if you spend a lot of time on Twitter, you'll appreciate its speed and efficiency.

Certain features aren't enabled by default, such as the ability to display inline images, as shown in our screenshot. This is really more of a novelty than anything else, as the images are inevitably low-resolution given the constraints of the terminal, but it's a lark nonetheless. Enter **config** to see a list of configuration options for *Rainbowstream*, and then enter:

```
config IMAGE_ON_TERM=True
```

Now, when you view the most recent tweets from a user (eg **view @linuxvoice**), linked images will be displayed inline.


mike@mike-megabox: ~
File Edit View Search Terminal Help
on? THERE ARE PRIZES GALORE
biglesp @biglesp 2014/10/28 10:44:19
RT @ben_nuttall: .@samaaron says "we're going to make a button that does X" should be "we're going to make a button - what do you want it to do?" #picademy
biglesp @biglesp 2014/10/28 10:54:05
RT @CodeClub: Hey #London peeps - want to help out @fth_nix with Scratch workshops for his Raspberry
am? https://www.eventbrite.co.uk/e/raspberry-jam-tickets-13296908411 http://t.co/NWyAFV03vg

If you guessed this is a Raspberry Pi, congratulations – you can now **apt-get remove** your X server.

PROJECT WEBSITE
www.rainbowstream.org

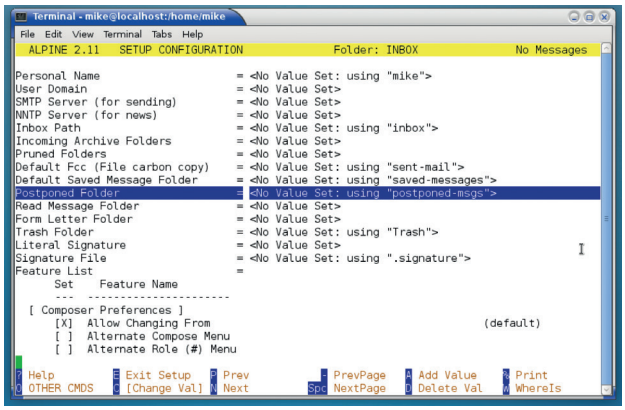
Mail client

Alpine 2.11

We've given plenty of kudos to the *Mutt* email client in previous issues of Linux Voice, so some readers have asked us to also give *Alpine* a mention too. And for good reason – it's a great program, and this author used it as his primary mail client for much of the early 2000s. To explain its history, however, we need to take a deep breath: *Alpine* is a continuation of *Pine*, an email client that started life in 1989 and was influenced by *Elm* (*Electronic Mail*) before it. However, *Alpine* development ceased in 2008, and since then users have made patches to add new features, along with a fork (*re-Alpine*).

Are you following? In practical terms, it doesn't really matter: if you install *Alpine* from your distro's package repositories, you'll almost certainly get an updated version with patches. When you start *Alpine*, you'll note that it's more welcoming than *Mutt*, with a menu-driven interface. However, like *Mutt*, there are keyboard shortcuts for various options – these are displayed along the bottom. If a shortcut looks like **^X**, that means you have to press Ctrl+X to activate it.

So, what makes *Alpine* great? Well, it's very fast to use: once you've learnt the keybindings, you can whizz around your mailboxes, reading, replying to and sorting mails in a fraction of the time it takes with the mouse. It's also highly customisable, and you can create colour themes for different types of messages. Almost all of the configuration can be achieved via the menus, so if you like the idea of a text-mode email client but don't want to spend hours hand-crafting configuration files, you'll love this.



Alpine is easy and quick to set up, and doesn't require extensive hacking of config files.

Alpine supports POP, IMAP and SMTP servers out of the box, and can even browse NNTP groups. It can render HTML emails, albeit without much formatting, and supports several mailbox formats.

"Alpine is a continuation of Pine, an email client that started life in 1989."

PROJECT WEBSITE
http://patches.freeiz.com/alpine

Source code analyser

PMD 5.2.0

Everyone makes mistakes, especially when you've been hacking away on the same piece of code for several hours. You may think you're working well, but niggling bugs could be creeping into your code. If you've got a deadline to reach and simply can't take a break, it's worth running a source code analyser on your work before you ship it.

PMD is one such analyser. It aims to find flaws in your code such as unnecessary object creation, unused variables, empty catch blocks and other bloopers. Currently it only works with Java, JavaScript, XML and XSL, but an additional tool called *CPD* (the "copy-paste detector") is provided, which identifies duplicated code in more languages including C, C++, C#, PHP and Ruby.

Both tools are written in Java, so extract **pmd-bin-5.2.0.zip** and jump into the **pmd-bin-5.2.0/bin/** directory. You'll need to provide a ruleset against which the code should be checked, along with a directory containing source code files. For instance, if you want to check some JavaScript in **/var/www/foo**, you'd use:

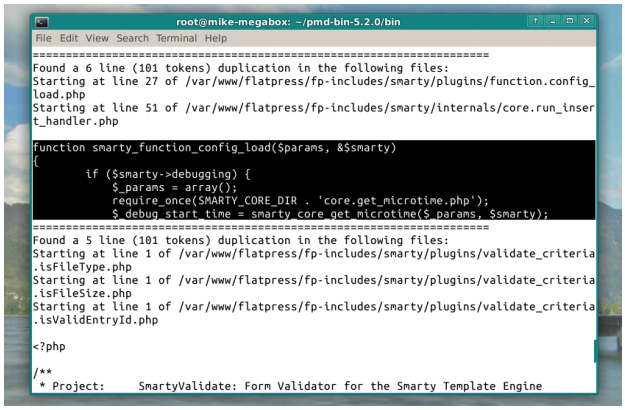
```
./run.sh pmd -R ecmascript-basic -d /var/www/foo
```

Many other rulesets are available – see the project's website for a full list. *PMD* will spit out its findings to **stdout**, or you can redirect them to another file by adding **> list.txt** to the end of the command.

To run *CPD*, you need to specify the language it should check, along with a location and the minimum token length, which should be reported as a duplicate. So you'd use something like this:

```
./run.sh cpd --minimum-tokens 100 --files /var/www/foo --language php
```

Ideally, this should help you to reduce duplication, and stop bugs from being repeated across other areas of the program. Plugins are available to integrate *PMD* and *CPD* with various IDEs, including *Eclipse*, *NetBeans* and *JBuilder*.



Here's *CPD* in action, spotting repeating sections of PHP code.

PROJECT WEBSITE
http://pmd.sourceforge.net

FOSSPICKS Brain Relaxers

Flight simulator

FlightGear 3.2

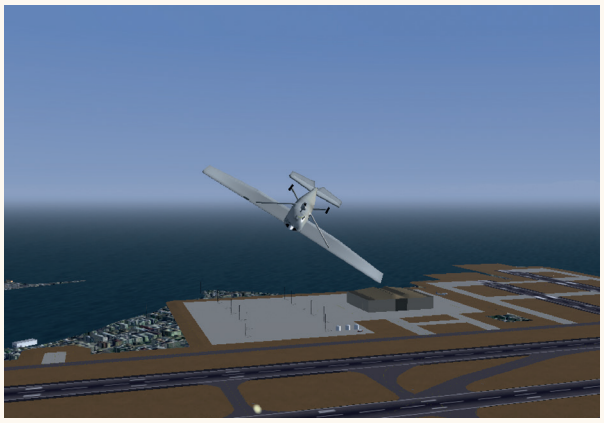
How many games can boast a 60,000 word manual? Fortunately, you don't need to read it all to play *FlightGear*, but if you want to get the most out of the simulator, you'll spend a lot of time with it. *FlightGear* is serious business: it has been in development for nearly 20 years, it's incredibly detailed, and it has realistic physics. Well, apart from the crashes...

It's also huge. You're looking at around 2GB for the complete game, which includes over 30 aircraft and hundreds of airports around the world. Getting it is a bit difficult, as there's no standard bundle for all Linux distros, so you're left to your own distro's repositories or third-party repos.

If you're running an Ubuntu variant, there's a PPA that has this latest *FlightGear* release (**ppa:saiarcot895/flightgear**).

If you've ever tried a complex flight sim before, you might've given up before even getting off the ground, thanks to the sheer number of controls involved. *FlightGear* is no different, but at least with the smaller aircraft like the Cessnas, you can enjoy some simple flying fun without burying your head in the docs. Hit S to start your engine, hold 9 to ramp up the throttle, and when you've got some speed, tap 8 to lift off. Then use the 0 and Enter keys to control the rudder.

You won't get far this way, as there are many more controls to master, but at least you'll have a bit of fun before crashing. (And really, the crashes aren't very realistic – you just bounce around on the ground.) Still, if you manage to take off and land a 777, write in and let us know...



"Hello, this is your captain speaking. We regret to announce some minor technical issues with this Ryanair flight to Málaga. Normal service will be resumed soon."

PROJECT WEBSITE
www.flightgear.org

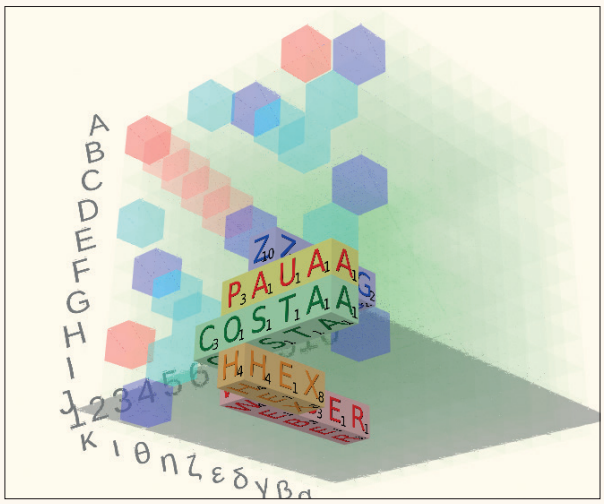
Multi-dimensional Scrabble game

Scrabble3D

Scrabble, like most popular board games, has been done to death now. Sure, it's still a great game, but from the squillions of computer versions we've seen over the years, there's little to differentiate them. Until we came across Scrabble3D, that is. This adds another dimension to the proceedings, and thereby a whole other level to the gameplay. The program is written using the *Lazarus* Pascal IDE, which means that packages are available with both *GTK* and *Qt* interfaces. The 64-bit Deb file worked fine on our Ubuntu 13.10 test machine; if you're running an RPM-based distro such as Fedora, you'll also find suitable packages at the project's website.

In its default mode, *Scrabble3D* lets you play a regular game with two, three or four players, dragging tiles from the top-right of the window onto the board on the left. The game asks for your language when you first start it, so that it can download the relevant dictionary, and you can change this later on. If you've got bored with the regular Scrabble rules, some alternatives are available: Clabbers (which lets you place anagrams of words) and Cambio Secco (whereby you can exchange all of your tiles, once per match, without losing your turn).

To find the 3D mode, go into Configuration > Settings, choose Advanced mode, go to Board > Configuration, and select 3D. After starting a new game, you can use the slider under the 3D view on the right to select a plane before dropping a tile. It's a bit fiddly at first, but the manual on the project's site explains it well.



Click Game > Run demo to see the CPU play against itself in the 3D mode.

PROJECT WEBSITE
http://scrabble.sourceforge.net